REMARKS/ARGUMENTS

The above-identified patent application has been amended and reconsideration and re-examination are hereby requested.

Claims 2-7 have been amended to correct errors pointed out by the Examiner.

Claims 1-5 stand rejected under 35 USC 102 as being anticipated by Close et al. (USP 6,125,466).

Applicant provides a method for determining Cyclic Redundancy Check (CRC) parity of data. The data is comprised of a plurality of bytes, each one of the bytes having a parity bit, the plurality of bytes of data having a CRC. The method as set forth in claim 1 includes generating the parity of the parity bits of the plurality of bytes of the data, such generated parity being the parity of the CRC of such data.

Close et al. describes a method for determining isolating the error in a byte by detecting whether there is a parity error in a row of region 24'; and by detecting whether bit there is a parity error in a column of region 24'. Applicant finds no description in Close et al., of generating the parity of the parity bits of the data. Applicant finds no description in Close et al., generating parity of the parity bits of the plurality of data bytes **_and_ _then comparing_** such **_generated parity_ _with_ _the parity bit of the CRC_** of the data. Applicant respectfully requests that the Examiner point to where Close et al. describes such comparison.

For example, referring to FIG. 1 of Close et al., assume there is an error in the third data down from the top of region 24' and in the fourth position from the left of region 24'. There will be a horizontal, X, or row, parity error in 22 of such third data down from the top of region 24'. This horizontal, X, or row parity error in 22 will not, by itself, be able to determine the vertical, Y, or row position in the having the error. Region 24" will however provide a parity error in third position from, Y, or column position but, by itself will not provide sufficient information to identify the horizontal, X, or row data having the error. Close et al., use both the horizontal parity of 22 in region 24' and the vertical parity in region

22" to identify the X,Y position of the error of the data in region 24'.

As noted above, Applicant generates the parity of the parity bits of the plurality of bytes of the data. Such step is not described in Close et al.

Reference is made to Close et al., at column 3, lines 41-46:

> Each byte 21 has a corresponding horizontal parity bit 22. ***For each column 28*** in which there is at least one byte of code stored, there is a corresponding vertical parity bit 27. ***These vertical parity bits 27*** make up the vertical parity database 24" which is located in a row 23 separate from rows 23 in the code space 24' in which runtime code is stored. (emphasis added)

Note that Close et al., do ***not*** say that they ***also*** calculate the vertical parity of each corresponding horizontal parity bit ***22, rather Close et al,*** ***only*** states that they calculate vertical parity over *each column 28 in which there is at least one byte of code stored.* Thus, while the parity bits in Close et al. are in the column 22 of region 24', as distinguished from the columns 28, Close et al, appear to use "***These vertical parity bits 27" and never describe using the parity bits in column 22 in region 24'.***

Further, refer to Close et al. column 3, lines 53-55:

> A vertical parity bit 27 *for each column* 28 containing code is generated when the code is compiled. (emphasis added)

Again, no mention of a vertical parity of the horizontal parity bits ***22***, as distinguished from ***28***.

Further, refer to Close et al. column 4, lines 10-13:

> A vertical parity check is then applied to the ***columns 28*** in the code space 24' of the addressed longword 29 to determine which bit of the byte(s) 21 contains the error, i.e., ***in which column 28,*** the error exists. emphasis added)

6

Again, no mention of the vertical parity of the horizontal parity bits *22*, as distinguished from *28*.

Claim 2 points out that Applicant provides a method for performing a check of the parity bit of a Cyclic Redundancy Check (CRC) of data, such data comprising a plurality of bytes, each byte having a parity bit. The method includes generating parity of the parity bits of the plurality of data bytes; *and comparing such generated parity with the parity bit of the CRC of the data.*

Close et al. do not generate parity of the parity bits of the plurality of data bytes; *and comparing such generated parity with the parity bit of the CRC of the data.*

Claim 3 points out that Applicant provides a method for determining Cyclic Redundancy Check (CRC) parity of data, such data having byte parity bits, the data having a CRC. *The method includes comparing the parity of the byte data parity with the parity bits of the CRC of the data.*

Close et al. do not *compare the parity of the byte dataparity with the parity bits of the CRC of the data.*

Claim 4 points out that Applicant provides a method for receiving data having a plurality of N bytes: [D(0), D(1), ... ,D(N-1)] each byte D(M) having a parity bit P(M); and *computing the parity of [P(0), P(1), ...P(N-1)].*

Close et al. do not *compute the parity of [P(0), P(1), ...P(N-1)].*

Claim 5 points out that Applicant provides a method for computing parity, p, of the Cycle Redundancy Check (CRC) of data protected with such (CRC). The method includes receiving data having a plurality of N bytes: [D(0), D(1), ... ,D(N-1)] each byte D(M) having a parity bit P(M); and *computing the parity of [P(0), P(1), ...P(N-1)]*, such computed parity being equal to the parity p of the CRC.

Close et al. do not *compute the parity of [P(0), P(1), ...P(N-1)]*

7

Application No.: 09/746,174
Response dated October 23, 2003
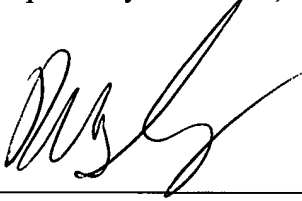Reply to Office Action of July 23, 2003

With regard to the Examiner's comment that claim 2 has the same coverage as claim 3 and claim 6 has the same coverage as claim 7, it is respectfully submitted that Applicant can claim the subject matter he/she regards as his/her invention in different ways and using different terms.

It appears that claim 7 is rejected only under 35 USC 112, second paragraph.

In the event any additional fee is required, please charge such amount to Patent and Trademark Office Deposit Account No. 50-0845.

Respectfully submitted,

_____10-23-03_____
Date

_____
Richard M. Sharkansky
Attorney for Applicant(s)
Reg. No.: 25,800
Daly, Crowley, & Mofford, LLP
275 Turnpike Street, Suite 101
Canton, MA 02021-2354
Telephone: (781) 401-9988, 23
Facsimile: (781) 401-9966

Q:\emc2\EMC2-085PUS\emc2-085pus rspns to oa due 10-23-03 28July03.cmcandrms_3.doc

8